AIAA Guidance, Navigation, and Control Conference and Exhibit
16 - 19 August 2004, Providence, Rhode Island

AIAA 2004-5228

Optimal arrival traffic spacing via dynamic programming*

Alexandre M. Bayen[†]
DGA - LRBA
Vernon, France

Todd Callantine[‡]
San Jose State University/NASA Ames Research Center
Moffett Field, CA

Claire J. Tomlin[§] Yinyu Ye[¶] Jiawei Zhang[||]
Stanford University
Palo Alto, CA

We present the application of dynamic programming to a combinatorial optimization problem to achieve proper arrival runway spacing, which appears in the process of assigning speed during the transition to approach and approach phases of flight. We apply the algorithm to data from a fast-time simulation developed under NASA's Advanced Air Transportation Technologies Project for investigating new air traffic management (ATM) concepts. For this research, the simulation is configured to simulate traffic inbound along two arrivals to the Dallas/Ft. Worth (DFW) airport, merging into a single stream at fix just prior to the final approach fix. We show how the algorithm computes the maximum minimum spacing between aircraft upon landing, and investigate the sensitivity of the spacing to perturbations.

Keywords: **Dynamic programming, fast-time simulation, arrival spacing.**

Introduction

The recent growth in embedded systems control applications has been triggered by numerous factors, which range from advances in monitoring technology, communications, and available computing power. It has provided engineers unprecedented capabilities for real-time control. Yet a bottleneck still exists in the design and realization of control laws for large scale systems, due to the complexity of the underlying mathematical problems. In particular, NP-completeness or non convexity of optimization problems cannot be eased by increases in computational power alone, but by new breakthroughs in algorithm design. Indeed, this type of difficulty does not preclude the tractability of solutions which are close to the optimal, fast to compute, and guaranteed to achieve a certain performance. Combinatorial optimization algorithms have proved to be very useful to solve this type of difficulties, and have emerged as a very powerful tool to solve planning problems for large scale systems, such as Air Traffic Control (ATC), or networks of multiple vehicles.

Several recent publications have used efficient algorithmic techniques to solve combinatorial optimization problems which appear naturally in large scale systems. In particular, for ATC, Ribichini and Frazzoli [10] reduce aircraft coordination problems to an integer program similar to facility location, for which they develop a decentralized approximation algorithm. Neogi [7] reduces the problem of landing aircraft assignment for multiple runway configurations to a Steiner Tree problem. Parallel landing with aircraft dependent separation requirement is known to be an NP-complete scheduling problem [8]. In [4], we develop a polynomial time algorithm to solve the problem of maximum spacing between aircraft when the set of feasible times of arrival is a single interval for every aircraft, and the landing order is not known a priori. We also developed an approximation algorithm for minimizing the latest landing time of a platoon of aircraft, the case in which the aircraft can hold for periodic time intervals [2]. This list of problems is not exhaustive, and becomes very large if Unmanned Aerial Vehicle (UAV) problems are included. In this paper, we will be interested in ATC applications.

The ATM system is a large-scale complex system that has evolved over decades to become highly ro-

*Research supported by NASA under Grant NCC 2-5422, by ONR under MURI contract N00014-02-1-0720, by DARPA under the Software Enabled Control Program (AFRL contract F33615-99-C-3014) and by a Graduate Fellowship of the Délégation Générale pour l'Armement (France).

†AIAA member, Major (Ingénieur Principal de l'Armement), DGA, France. Autonomous Navigation Laboratory, LRBA, Vernon, France. Corresponding author. bayen@stanford.edu. Tel: (33-2)-27.24.41.87, Fax: (33-2)-27.24.44.99

‡AIAA member, Senior Research Engineer, Human Factors Research and Technology.

§AIAA member, Assistant Professor, Aeronautics and Astronautics, and, by courtesy, Electrical Engineering.

¶Professor, Management Sciences and Engineering.

||Ph.D. Student, Management Sciences and Engineering.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**
4. TITLE AND SUBTITLE **Optimal arrival traffic spacing via dynamic programming**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stanford University,Department of Aeronautics and Astronautics,Stanford,CA,94305**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

bust and safe. It includes both human agents, such as pilots, air traffic service providers (ATSPs), and airline operations center (AOC) personnel, and machine agents (e.g., aircraft flight management systems (FMSs) and ATC automation). The present-day ATM system remains vulnerable to environmental disturbances and unpredictable in many respects. It relies on conservative margins and closely monitored tactical operations to ensure safety, which in turn leads to inefficiencies.

Researchers seek a future ATM system that leverages advances in aircraft automation, ATC automation, and communications, navigation, and surveillance systems to decrease delays, noise, and fuel usage while improving safety. One class of proposed future ATM concepts entails keeping aircraft on FMS routes until stabilized on final approach, enabling them to fly more precise and efficient trajectories in the terminal (TRACON) airspace. For example, FMS procedures enable 'continuous descent approaches (CDAs),' which afford greater fuel efficiency and reduced noise if they can be flown safely.

Efficient TRACON ATM rests on the ATSP's ability to predict future aircraft locations and issue clearances appropriate for merging aircraft and maintaining required separation and flow spacing. TRACON ATSPs use speed changes and lateral maneuvers to control traffic. Control options are limited by both the area available for maneuvering and the overriding concern of pilots to stabilize the aircraft for the approach. Accelerations are therefore not likely to be desirable to pilots, while lateral maneuvers may lead to adjusting the overall arrival flow if traffic density is sufficiently high.

Studies in today's ATM environment have shown that ATSPs find it difficult to predict the 4D trajectory of aircraft flying CDAs. ATSPs therefore add a large buffer to ensure safety, which dramatically limits throughput [7]. One challenge is therefore to develop an ATM concept that yields CDA-type benefits with moderate to high throughput. ATC automation could potentially support ATSPs toward this end. For example, automation tools could use 4D FMS trajectories downlinked by aircraft to provide ATSPs with advisories on suitable speed changes to issue aircraft to null schedule errors created by environmental factors. In this paper, we address issues surrounding speed advisories used in conjunction with scheduling automation to reduce excess arrival spacing buffers.

Investigating new ATM concepts is inherently difficult due to the number of agents and overall complexity of the system. Human agents are essential for system robustness, and acceptability of procedures and automation tools to human agents is crucial. A promising concept must therefore undergo extensive human-in-the-loop simulation to refine these elements. However, a number of human factors issues may be ex-

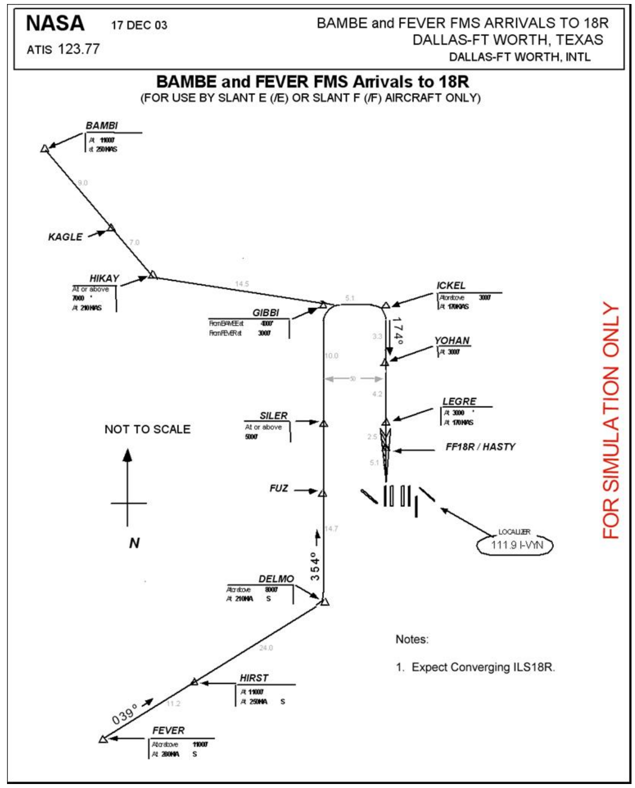

Fig. 1 Arrival chart showing merging FMS routes. Aircraft are expected to fly the arrival in LNAV/VNAV until glideslope intercept.

amined using simulations with embedded agent models. These include the capability of ATC automation to generate usable advisories, the pace at which ATSPs must issue various types of clearances to control traffic under various conditions, the types and magnitudes of disturbances that ATSPs can manage given proposed routings and clearances, and related issues. In this research we use a Java$^{\mathrm{TM}}$-based fast-time simulation tool called TCSim (Trajectory-Centered Simulation) to provide relevant data and visualize proposed operational concepts.

This paper is organized as follows. We first describe the capabilities of TCSim and how we use it to simulate a specific arrival problem at the DFW airport. Then, we describe combinatorial optimization problems which appear in TCSim, and derive a polynomial time algorithm to solve it, using dynamic programming. We finally show results of simulations run for the DFW airport, which illustrate the use of TCSim to study the sensitivity of spacing to disturbances.

Spacing arrival traffic

TCSim was developed at NASA to provide capabilities useful for investigating 'trajectory-oriented' ATM concepts. We are using it here to investigate a problem that involves merging arrival flows in the DFW TRACON (see Figure 1). In this configuration, northwest and southwest arrivals enter the TRACON over the BAMBE and FEVER meter fixes, respectively,

merge at GIBBI, and land on runway 18R spaced according to a standard wake vortex spacing matrix. This problem involves several issues, including suitable FMS routes that aircraft can fly using typical Lateral/Vertical Navigation (LNAV/VNAV) modes, aircraft scheduling at the meter fixes and runway, and the design of control strategies and supporting ground automation that can null schedule errors to produce an efficient arrival flow. Variations and extensions for future study include the simultaneous use of runway 13R, aircraft equipped to meet required-times-of-arrival (RTAs) assigned by ATSPs, and relative self-spacing guidance.

TCSim is designed to simulate common types of commercial aircraft flying FMS trajectories in fast time. It can use the trajectories it generates for each aircraft to emulate complex air- and ground automation functionality. TCSim also includes embedded ATSP and pilot agents for simulating interactions involved with specific concepts and traffic conditions. TCSim is capable of automatically generating traffic scenarios with specific characteristics to support Monte Carlo studies, and of producing a variety of metrics for each trial. Researchers can specify charted routes, traffic characteristics, agent behavior, and structure of simulation trials to conduct via a configuration file.

TCSim represents trajectories as a sequence of lateral legs (including turn segments), and associated 'vertical' legs that specify when computed altitude and speed rates of change are in force. A rule set guides how TCSim constructs vertical trajectory segments. TCSim produces vertical trajectories that map closely to observed VNAV profiles. Figure 2 shows TCSim vertical segment types, and how they might appear in an aircraft's approach and landing trajectory. Working backwards from touchdown, each aircraft has a segment that represents a stable landing configuration at a landing speed appropriate for the aircraft's type. Prior to stabilization, other legs along the glideslope represent decelerations for successive flap extensions. The glideslope intercept point depends on the crossing restriction at the final approach fix; aircraft with high landing speeds may extend flaps earlier than shown in Figure 2. TCSim constructs trajectory segments that honor the type (i.e., at, at-or-above, or at-or-below) and values for speed and altitude restrictions. Decelerations to meet crossing speeds use a deceleration segment with a two degree flight path angle in the absence of other constraints; TCSim can also model steeper speedbrake-assisted decelerations. Crossing restriction locations dictate the need for trajectory segments with computed flight path angles. It is important that charted speed and altitude restrictions are designed so that VNAV can reasonably fly them in the given wind field; otherwise TCSim will construct unrealistic trajectories. Figure 3 plots the actual trajectories TCSim produces for aircraft flying the charted FMS arrival routes shown in Figure 1.

Once TCSim computes a trajectory for a particular aircraft, it 'flies' the trajectory in the background to collect accurate estimated times-of-arrival (ETAs) at each lateral point along the trajectory. These estimates support emulation of automatically downlinked trajectory information for input to ground-based scheduling automation. TCSim can also evaluate potential changes to the trajectory and supply ETAs for aircraft given the changes. This capability is essential to the present research; we use TCSim to output a set of feasible arrival times for each arriving aircraft that serve as input to the dynamic programming algorithm.

TCSim also contains embedded ATSP agents. Simulated ATSPs can issue a variety of clearances to aircraft, including heading, altitude and speed clearances, as well as more complex clearances with crossing restrictions, and a variety of execution conditions. Regardless of clearance type, the aircraft's flight path is still represented as a trajectory. Pilot agents are currently modeled by the time it takes and the method used to implement a particular clearance (TCSim may also simulate flight technical errors, including clearance non-compliance). For example, the new trajectory for an aircraft cleared to fly a particular course contains a short delay leg at the beginning to represent the aircraft's trajectory during the time the crew takes to begin the turn. After the delay leg the trajectory includes a turn segment and, finally, a straight segment on the specified course. Speed clearances operate similarly; they may include a delay segment that represents the time it takes a crew to configure the aircraft automation or flight surfaces. Speed clearances that effectively cancel one or more speed restrictions entail backwards computation of the trajectory without the cancelled restrictions, as well as forward integration through the wind field to construct the required speed change segments.

Each ATSP agent uses a particular 'control strategy' that specifies what ATC automation is available to them and when they address each aircraft. No attempt is made to simulate the actual process of controlling air traffic, as in other research [5, 9]. Instead, each controller addresses an arriving aircraft as it passes specified 'control points' (in general, the farther a control point is from the runway, the more potential control authority it offers). The ATSP agents consult simulated ATC automation to produce specified advisories for that aircraft. When actively controlling traffic, agents can select an advisory to null each aircraft's arrival schedule error, and issue it as a clearance.

TCSim ATSP agents have been used, for example, to show that a single speed advisory clearance can null sufficiently small schedule errors caused by

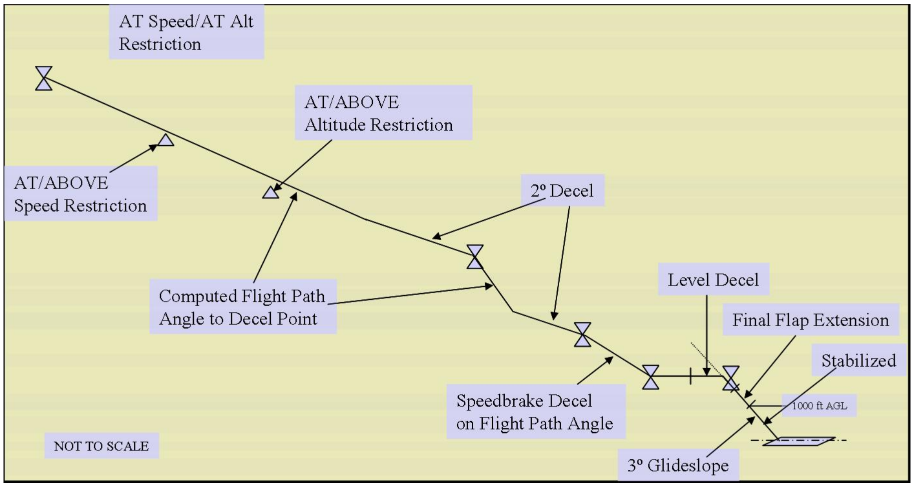

Fig. 2 Generic vertical trajectory illustrating vertical segment types used in TCSim.

errors in meter fix crossing (TRACON entry) times and predicted landing speeds [6]. In that research, the TRACON merge problem was investigated Monte Carlo-style as follows. TCSim automatically generated arrival flows across both meter fixes to runway 18R. It first generated a specified number of aircraft of types selected to represent a typical DFW arrival flow. It assigned each aircraft to arrive from a given direction (based on a specified minimum number that must come from each direction), and generated a trajectory corresponding to the appropriate FMS route. TCSim then established simulation entry times for each aircraft that simulated a coordinated schedule for flows across both meter fixes to yield proper wake vortex spacing at the runway threshold. It then adjusted the entry times probabilistically to produce meter fix crossing time errors, and adjusted the landing speeds and recomputed the trajectory to simulate predicted landing speed errors.

With the scenario specified, TCSim began simulating the traffic in real-time. ATSP agents issued automatically generated advisories at specified control points in an attempt to null any schedule deviation for each aircraft. We specified that only decelerations could be issued, on the assumption that pilots are not likely to find accelerations along the transition to final approach desirable. On successive trials with each traffic scenario, TCSim incremented an 'excess spacing buffer' that was added during the scheduling process. When the buffer was small, separation violations at the merge point (GIBBI) and subsequent locations were observed. The results of multiple trials yielded (among other useful measures) the excess spacing buffer required to ensure safe operations for the given traffic scenario under the given control strategy with disturbances of particular types sampled from specific distributions. Results of this sort not only indicate the effectiveness of a given control strategy, but also inform the development of automation tools, traffic scenarios, clearance phraseology, and other elements required to conduct useful human-in-the-loop studies of future ATM concepts.

In the research reported here, we use TCSim to provide a set of feasible times of arrival for each aircraft. The times are represented in seconds from a global reference (elapsed seconds from simulation start). The arrival times reflect when each aircraft will arrive given its current trajectory, as well as when it would arrive if issued a range of speed clearances in five knot decrements down to the final approach fix crossing speed. The algorithm presented below can use this information to automatically space arrival traffic. The set of feasible arrival times provides the possibility of optimizing the spacing according to a user-defined cost (which is arbitrary) while maintaining safety. For the present paper, we will maximize the minimum time separation between successive landings.

A polynomial time algorithm for maximal spacing

The problem of computing the maximal minimum spacing of jobs scheduled in a single processor, which we investigate in this paper, is not a standard problem in scheduling. Usual scheduling problem in the literature try to minimize a cost relevant for computer science or operations research applications, for example the sum of all processing times, the makespan (time at which the last job is finished), and the sum of the delays [1]. In the present context, we are interested in maximizing the separation between landing times, given hard constraints which define the set of possible arrival times. This problem was posed in its general form as a mixed integer linear program (MILP) in [3], and was solved for specific cases in [4, 2]. In this article, we investigate a subcase of the general problem which appeared while using TCSim, which is a purely discrete version of the general MILP [1] in [3].

[1]Note that in general, nothing enables to know a priori if the transformation from a MILP into an integer program (IP) makes it easier or harder to solve.

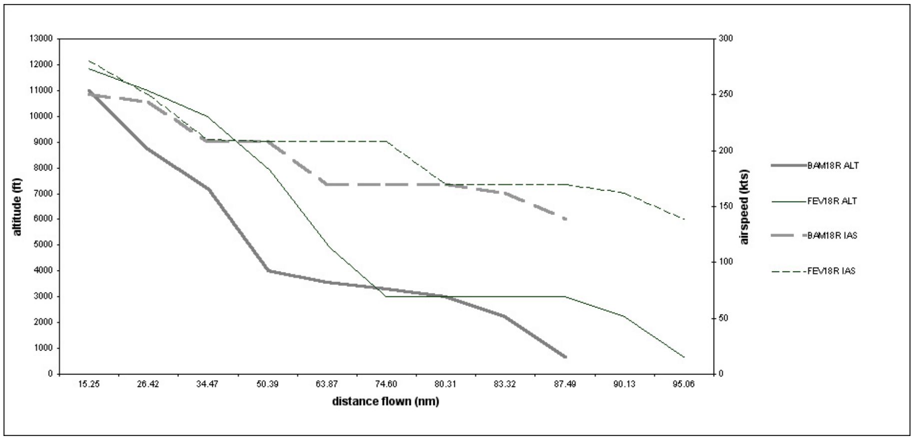

Fig. 3 TCSim altitude and speed profiles for aircraft flying the BAMBE and FEVER FMS arrivals to DFW runway 18R.

Problem formulation

Consider N aircraft labelled by the index $i \in \{1, \cdots, N\}$. The arrival time τ_i of aircraft i belongs to the a set of n_i feasible arrival times (computed by TCSim), called $t_{i,j}$, indexed by $j \in \{1, \cdots, n_i\}$. In most of the cases run in TCSim, the following condition holds: for all i and i' such that $i > i'$, $t_{i,j} > t_{i',j'}$ for all j and j'. This means that no matter what choice the algorithm makes, aircraft always arrive in increasing order of i. This is the case whenever the perturbations from the schedule TCSim uses to generate the traffic scenario are sufficiently small, and the control points and speed restrictions along the arrival route are selected so the speeds fall in a reasonably narrow range. (The TCSim scenarios used here reflect these criteria: perturbations do not exceed approximately thirty seconds, and the available speed range is roughly forty knots.) While ATSPs could conceivably issue clearances (e.g., heading vectors) that effectively resequence the arrivals, the simulated arrival concept requires suitably small deviations, because it specifically examines the potential for using speed clearances alone to properly space the aircraft. For this paper, we want to find the maximum minimum spacing between the aircraft at the destination airport, mathematically defined by the following optimization program:

$$\begin{array}{lll} \textbf{max:} & \Delta & \\ \textbf{s.t.:} & \tau_i \in \{t_{i,j}\}_{j\in\{1,\cdots,n_i\}} & \forall i \in \{1,\cdots,N\} \\ & \tau_i - \tau_{i-1} \geq \Delta & \forall i \in \{2,\cdots,N\} \end{array} \quad (1)$$

In the previous program, Δ represents the maximum minimum spacing between aircraft upon landing; the $t_{i,j} \in \mathbf{R}$ are given (computed by TCSim). This program is a fully discrete version of the MILP solved in polynomial time in [4]. The optimization program solved in [4] is obtained from (1) by allowing any arrival time between the earliest and the latest:

$$\tau_i \in [\min_{j=1}^{n_i} t_{i,j}, \max_{j=1}^{n_i} t_{i,j}]$$

and allowing any order of arrival of the aircraft, i.e.:

$$|\tau_i - \tau_{i-1}| \geq \Delta$$

Dynamic programming algorithm

We now show how to solve this problem exactly using dynamic programming.

Proposition 1. *The solution Δ to the optimization program (1) is given by $\Delta = \delta(N, n_N)$, where, $\delta(N, n_N)$ is defined by the following recurrence:*

$$\begin{aligned} \delta(2,j) &= t_{2,j} - t_{1,1} \\ \delta(i,j) &= \max_{j'=1}^{n_{i-1}} \{\min\{t_{i,j} - t_{i-1,j'} \ , \ \delta(i-1,j')\}\} \end{aligned} \quad (2)$$

The computational complexity of the algorithm is $O(TM)$ where $T = \sum_{i=1}^{N} n_i$ is the total number of arrival time points, and $M = \max_{i=1}^{N} n_i$.

Proof. For all $j \in \{1, \cdots, n_i\}$, $\delta(2, j)$ is by construction the largest separation between aircraft 1 and 2 if aircraft 2 arrives at $t_{2,j}$, since $t_{1,1}$ is the earliest possible arrival time of aircraft 1.

Suppose that $\delta(i-1, j')$ represents the largest minimum separation between consecutive aircraft up to $i-1$, if aircraft $i-1$ arrives at time $t_{i-1,j'}$. Then, $\min\{t_{i,j} - t_{i-1,j'} \ , \ \delta(i-1,j')\}$ is the minimum between two quantities: (ι) the separation $t_{i,j} - t_{i-1,j'}$ between aircraft i and aircraft $i-1$ if aircraft i is scheduled at $t_{i,j}$ and aircraft $i-1$ at $t_{i-1,j'}$; and $(\iota\iota)$ the minimum separation $\delta(i-1, j')$ between consecutive aircraft up to $i-1$, if aircraft $i-1$ arrives at $t_{i-1,j'}$. The minimum of (ι) and $(\iota\iota)$ is the minimum separation between consecutive aircraft up to i. Taking the maximum over the set $\{1, \cdots, n_i\}$ provides the largest minimum separation between the i consecutive aircraft.

The best schedule is obtained if the last aircraft arrives as late as possible, i.e. if $\tau_N = t_{N,n_N}$. This number is $\delta(N, n_N)$

There are T evaluations of $\delta(\cdot,\cdot)$ to do. Each of them requires taking the maximum between M numbers. Therefore, the complexity is $O(TM)$. □

Corollary 2. *The previous result holds even when the arrival times overlap, provided the recurrence is replaced by:*

$$\delta(i,j) = \max_{\substack{j' \in \{1,\cdots,n_{i-1}\} \\ t_{i-1,j'} \leq t_{i,j}}} \{\min\{t_{i,j} - t_{i-1,j'} \ , \ \delta(i-1,j')\}\}$$

Proof. The proof is the same as for Proposition 1. For a given choice of $t_{i,j}$ for aircraft i, the set

$$\{j' \mid j' \in \{1,\cdots,n_{i-1}\}\,,\ t_{i-1,j'} \leq t_{i,j}\}$$

represents the set of available arrival times of aircraft $i-1$ which precede $t_{i,j}$, and therefore respect the order of arrival of the aircraft. □

Remark. The computation $\delta(i,j)$ of (2) can be done in $\log_2(M)$ time using a bisection method on $j'=1$ to n_{i-1}. This is because $t_{i,j} - t_{i-1,j'}$ is decreasing in j' and $\delta(i-1,j')$ is increasing in j', so that the maximum is achieved when the two are crossed over. This can be done using bisection on n_{i-1} possible points. Therefore, the complexity of the algorithm is $O(T\log_2(M))$. Note that M is small in practice; therefore, this observation will not be used in the implementation.

The present algorithm requires arrival times to be known a priori. The problem of unknown arrival order might conceivably be addressed through a combination of trajectory-based prediction and downlinked wind updates, so that earlier arrivals enable the algorithm to be applied to later arrivals. Applications of this sort are beyond the scope of this paper. We believe that a generalization of Baptiste's algorithm [1] would solve this problem in polynomial time (i.e. even if the order of aircraft is not fixed anymore).

Application to fast-time simulation data

Application to fast-time simulation data. The algorithm was applied to TCSim data for the merging flows at DFW. We used it for two purposes. First, in order to find the maximum minimum spacing upon landing under uncertainty. Second, in order to show how to use this tool to quantify the perturbation threshold above which it is not possible anymore to guarantee a given minimum spacing upon landing. Typically, the type of data generated by TCSim reads as follows:

```
ATA001 1295 1305 1310 1311 1316 1320 1325 1327 1335
UAL002 1413 1423 1429 1439 1447 1458 1468 1478 1489
DAL003 1522 1532 1541 1551 1557 1567 1581 1587 1601
UAL004 1606 1613 1619 1629 1638 1648 1659 1673 1682
COA005 1693 1700 1705 1710 1710 1715 1720 1725 1730
SWA006 1787 1794 1799 1799 1804 1809 1814 1819 1824
```

where the names in the first column are dummy, and the arrival time is in seconds, measured from an arbitrary reference. We realized a set of simulations involving 20 aircraft of the same type. This assumption enables us to ignore differences in required wake-vortex spacing distances due to different aircraft types.

To produce the data, TCSim performs the process described above. It formulates each arrival traffic scenario based on a schedule, but produces estimated arrival times that reflect TRACON-entry time and predicted landing speed perturbations. As aircraft arriving from the north cross HIKAY, and southern arrivals cross DELMO, TCSim computes the set of feasible arrival times. For the application concept (i.e., speed adjustments alone), we assume that by this point the ATSPs will have assigned the landing order for the merged flow. However, the exact landing time has not been assigned.

Maximum minimum spacing upon landing

In order to illustrate the variation of the maximum minimum spacing with the number of aircraft which have already crossed this point (at which ATC would issue the corresponding speed assignment), we compute its average value over 20 runs. In the final version of the paper, we will display the result of a Monte-Carlo simulation. Figure 5 shows the variation of Δ with the number n of aircraft which have crossed the metering point. As expected, this minimum separation decreases with the number of aircraft (the more aircraft are taken by the algorithm, the more the margin for adjustment decreases). The asymptote is around 90 seconds, which is around the value expected for this type of data set.

Figure 6 illustrates six scenarios for which we have compared the maximum minimum spacing provided by the algorithm and the spacing provided by TCSim. By construction, the algorithm does not take other factors than time spacing into account, which explains the differences between the two results, illustrated in Figure 6. Several interesting results appear from these simulations:

- An appearance of oscillations is observed in the TCSim data. This is in fact due to the random nature of the perturbations from TCSim schedule, coupled with the fact that the simulated ATSP control strategy only allows decelerations. Moreover, for this research, the ATSP agents control every aircraft to the original schedule, including the first aircraft. This means that if the first aircraft happens to be ahead of schedule, it will nonetheless be slowed down (which costs valuable space, but recognizes that another aircraft might actually be just ahead of the simulated flow). Inter-arrival spacing dips whenever an aircraft that happens to be behind the TCSim schedule is followed by one that is ahead, because the ATSP

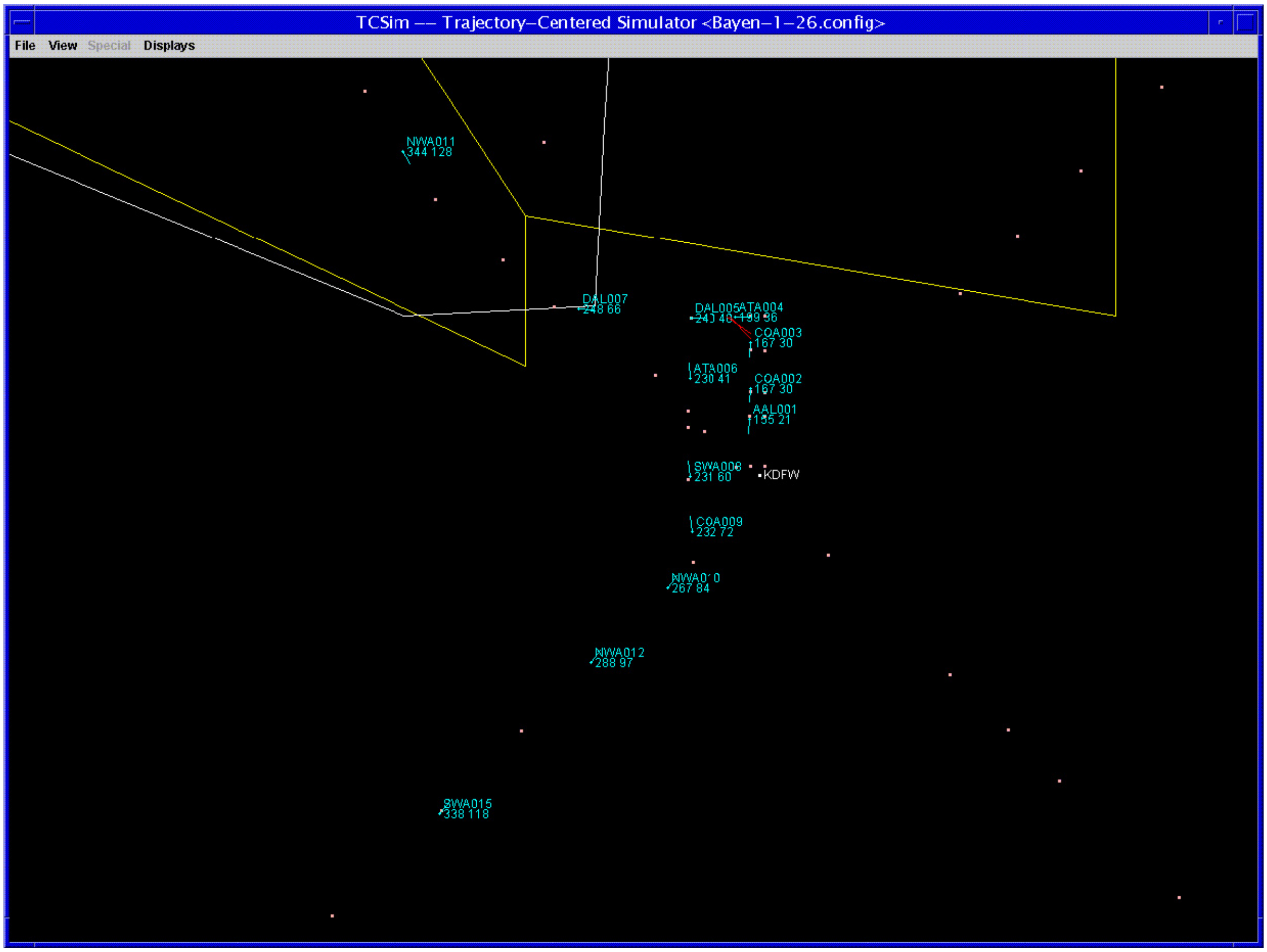

Fig. 4 Screen snapshot of TCSim simulating merging arrival flows along the charted arrival routes shown in Figure

cannot accelerate the lead aircraft, and may only slow the trail aircraft to its originally scheduled arrival time.

- Despite differences in schedules, if TCSim and the algorithm happen to agree for the schedule of two successive aircraft, it sometimes starts a new sequence of aircraft for which the two methods agree, which is intuitive, since they both try to assign as little delay as possible, once the constraints are met.

The results indicate that the optimal spacing algorithm can provide benefits if some conditions are met. The algorithm would then produce an optimal new schedule for an arrival flow. First, a suitable mechanism for generating feasible arrival times under specific conditions must be available. This is an instance of a class of problems applicable to numerous future ATM concepts. Second, the arrival flow must be reasonably well established, and the points at which the arrival times are generated must be located such that the times become available to the algorithm in sequence. This means that the ATSP can immediately issue the advised speed to the aircraft. Relaxing this requirement may be possible by, for example, repeating the 'speed adjustment' process. Further research with TCSim will explore these and related issues.

Threshold perturbation intensity

The results presented above also illustrate the functionality of TCSim for which we want to use this algorithm: for a given perturbation in the schedule (due to weather or other factors), which can be simulated by TCSim, the algorithm enables the computation of the worst case scenario under the optimal action of ATC. In future implementations, this functionality will thus enable us to compute the perturbation threshold above which we cannot guarantee a given spacing requirement at the destination airport. In particular, we are interested in running a Monte Carlo simulation in order to compute the threshold covariance of the perturbation above which airport given specifications become impossible to meet.

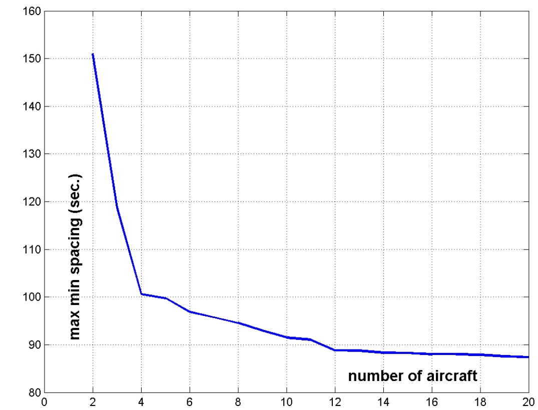

Fig. 5 Variation of the maximum minimum time separation Δ with the number of aircraft which have crossed the metering fix. This value is obtained by averaging 20 runs and will be obtained by a Monte Carlo simulation in the final version of the paper.

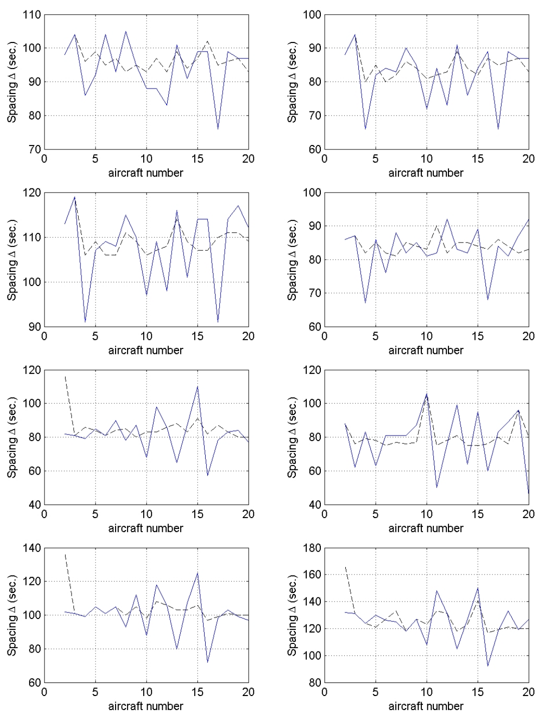

Fig. 6 Example of spacing computation of TCSim (solid curve) and the dynamic programming algorithm (dashed). Each point of the curve represent the Δ computed by both methods, for two successive aircraft.

Conclusion

We have presented some of the simulation capability of TCSim, and applied it to the Dallas/Ft. Worth airport airport. As a building block of our method, we have derived a simple dynamic programming based algorithm to solve for the maximum minimum spacing between aircraft upon landing. The results of this algorithm were compared to results of TCSim, which are not only based on spacing, but take into account other factors related to ATC. We have shown how these results will be used in future to characterize perturbation thresholds above which airport spacing constraints cannot be met by Air Traffic Control

Acknowledgments

The authors are grateful to Zoë Abrams for fruitful discussions leading to the dynamic programming algorithm presented in the paper.

References

[1] P. Baptiste. Polynomial time algorithms for minimizing the weighted number of late jobs on a single machine when processing times are equal. *Journal of Scheduling*, 2:245–252, 1999.

[2] A. M. Bayen. *Computational Control of Networks of Dynamical Systems: applications to the National Airspace System.* PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 2003.

[3] A. M. Bayen and C. J. Tomlin. Real-time discrete control law synthesis for hybrid systems using MILP: applications to congested airspaces. In *Proceedings of the 2003 American Control Conference*, Denver, CO, May 2003.

[4] A. M. Bayen, C. J. Tomlin, Y. Ye, and J. Zhang. Polynomial time algorithm for a MILP formulation of an aircraft scheduling problem. In *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003.

[5] T. Callantine. CATS-based air traffic controller agents. Technical Report NASA Contractor Report 2002-211856, NASA Ames Research Center, Moffett Field, CA, 2002.

[6] T. Callantine and E. Palmer. Fast-time simulation studies of terminal-area spacing and merging concepts. In *Proceedings of the 22nd Digital Avionics Systems Conference*, 2003. Ref: 03CH37449.

[7] N. Neogi. Aircraft assignment for multiple runway configurations. Submitted to the *2004 AIAA Conference on Guidance, Navigation and Control*, Jan. 2004.

[8] F. Neuman and H. Erzberger. Analysis of sequencing and scheduling methods for arrival traffic. Technical report, NASA Ames Research Center, Moffett Field, CA, NASA-TM-102795 1990.

[9] A. Pritchett, S. Lee, M. Abkin, A. Gilgur, R. Bea, K. Corker, S. Verma, A. Jadhav, T. Reynolds, L. Vigeant-Langlois, and G. Gosling. Examining air transportation safety issues through agent-based simulation incorporating human performance models. In *Proceedings of the* 21^{st} *Digital Avionics Systems Conference*, Irvine, CA, Oct. 2002. Ref: A.5-1-7.A.5-13.

[10] G. Ribichini and E. Frazzoli. Efficient coordination of multiple-aircraft systems. In *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003.